STAT 5291 Final Project

Air Pollution Prediction in Beijing

Prof David Rios

Xinming Pan (xp2203)
Yige Yang (yy3266)
Peiyu Yang (py2300)
Yiyi Zhu (yz4370)
Tianhong Chen (tc3234)

Date: April 24, 2023

Abstract

This project utilizes daily air pollution data from Beijing between 2010 and 2014, in combination with natural factors that may influence air quality, such as snowfall, rainfall, wind speed, dew point, and temperature, to investigate the patterns of Air Quality Index (AQI) fluctuations (Chen, 2017). A higher AQI value indicates more severe air pollution. After conducting exploratory data analysis, we gained a preliminary understanding of the data characteristics and presented some related features through charts. In this project, we experimented with parametric and non-parametric models, including linear regression, ARIMA model, SARIMA model, and LSTM models. The results show that the LSTM model demonstrates higher accuracy and stability in predicting AQI.

1. Introduction

In the years leading up to the 2008 Beijing Summer Olympics, air pollution in China continued to worsen. According to statistics, approximately 10% of haze was caused by natural factors, while nearly 90% of pollution was attributed to human emissions. Industrial activities such as coal burning, chemical production, and heavy metal smelting were the primary causes of air pollution in the surrounding areas of Beijing. Air pollution has serious negative impacts on climate, environment, and health. The presence of PM2.5 and other pollutants not only increases the mortality rate of people with serious and chronic illnesses but also exacerbates respiratory and cardiovascular diseases, alters lung function and structure, and affects the human immune system. To address this problem, the government implemented measures to reduce air pollution in Beijing and its surrounding cities.

In this project, we propose a data mining approach to analyze the air pollution situation in Beijing, as well as the influence of natural factors on air pollution, from two years after the 2008 Olympics (2010) to 2014. We hope we can get an insight into what factors air pollution depends on, and determine whether it has some trends or patterns.

2. Data Source & Description

2.1 Data Introduction

The data contains hourly PM2.5 data of the US Embassy in Beijing. Meanwhile, meteorological data from Beijing Capital International Airport are also included (Chen, 2017). Then, we combine it daily (Huertas, 2022). The data presents the daily air pollution index and the index for the previous day in Beijing from 2010 to 2014, along with six natural factors that may potentially influence the air pollution index: wind speed, snow, rain, dew, temperature, and press. Each observation has 8 numerical variables, including 7 independent variables and one dependent variable. The air quality index (AQI) is classified into different categories based on the pollution levels. An AQI of 0-50 represents good air quality, while a value between 51-100 is considered moderate. An AQI above 100 is deemed unhealthy for sensitive groups.

2.2 Data processing

To ensure the accuracy of the data analysis, we first need to check whether there are any missing values in the dataset and remove observations containing missing values. In this dataset, we found no missing values. Secondly, we need to adjust the date format to the year-month-day form and confirm that the data in the other eight columns is in numerical format.

2.3 Exploratory Data Analysis

2.3.1 Correlation matrix

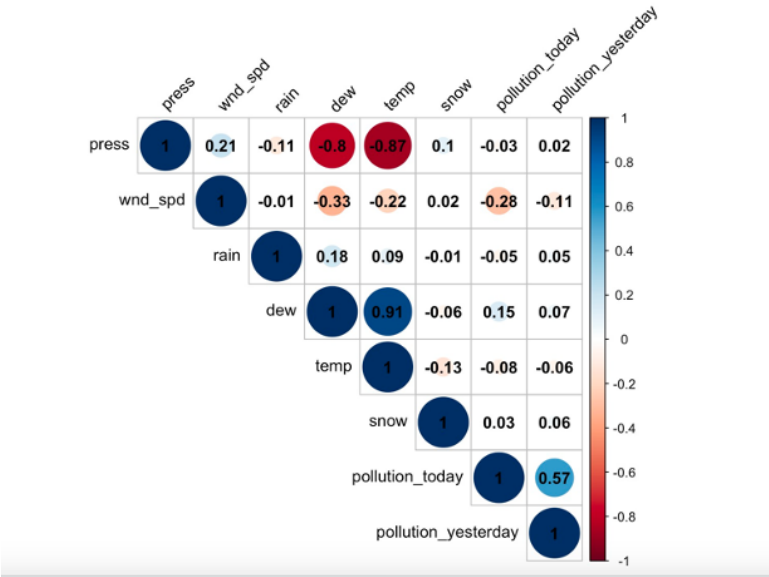


Figure 2.1

From the correlation matrix in Figure 2.1, there is a strong relationship between pollution_today and pollution_yesterday, with a correlation coefficient of 0.57. This suggests that a high AQI from the previous day is likely to result in a high AQI today. The second strongest positive correlation, with a coefficient of 0.15, is between dew and pollution_today, indicating that the AQI is likely to be high when the dew point is high. There is also a strong negative relationship between wind speed and pollution_today, which is -0.28, which demonstrates that a high AQI tends to coincide with slower wind speeds.

2.3.2 Line plots

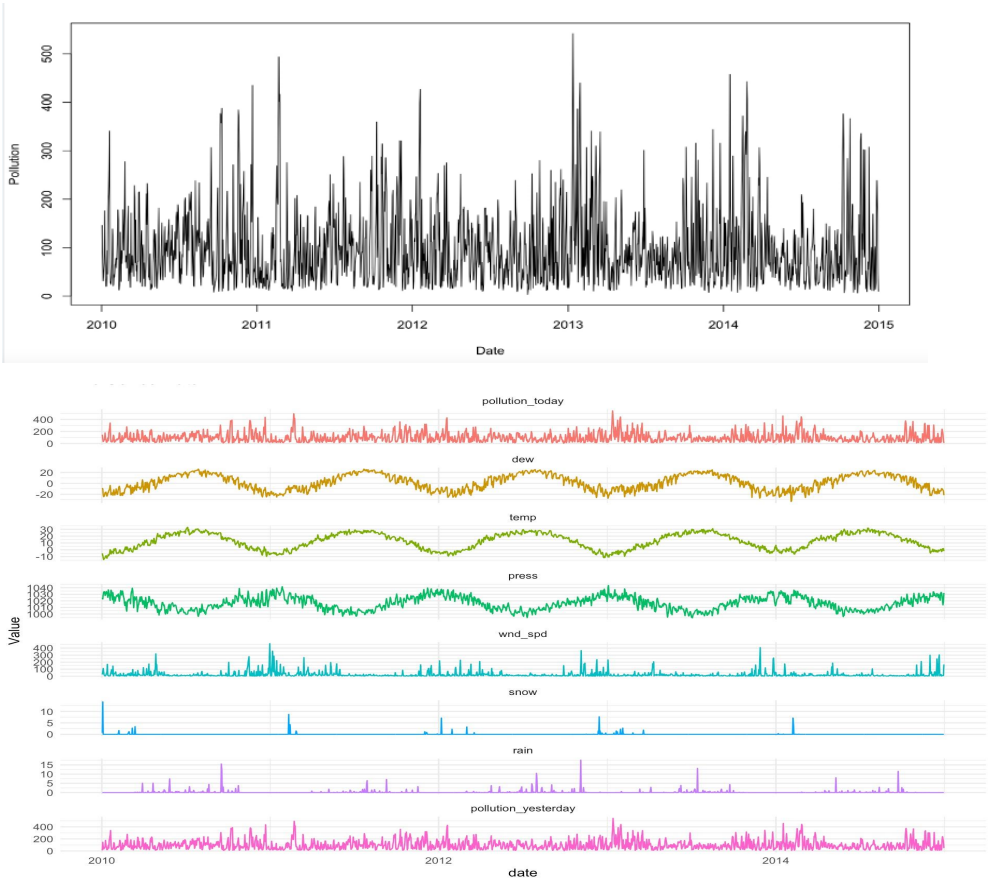
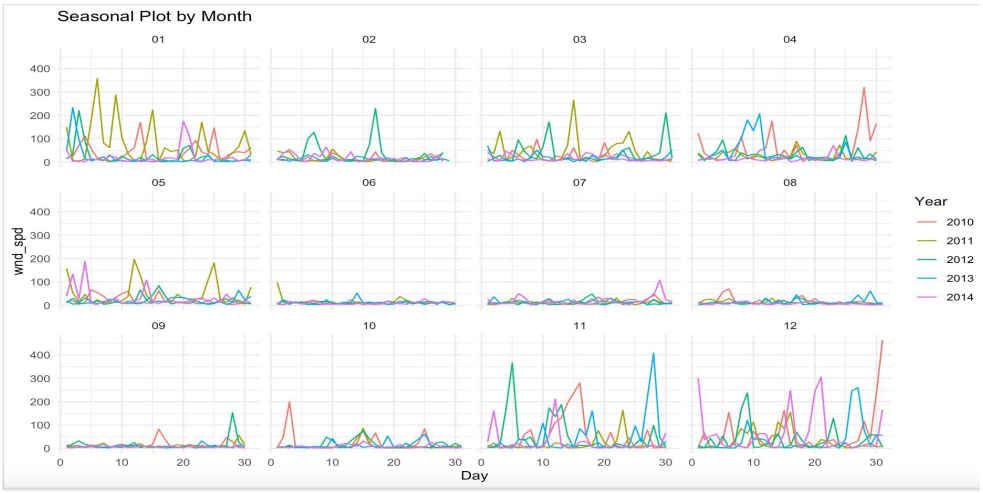
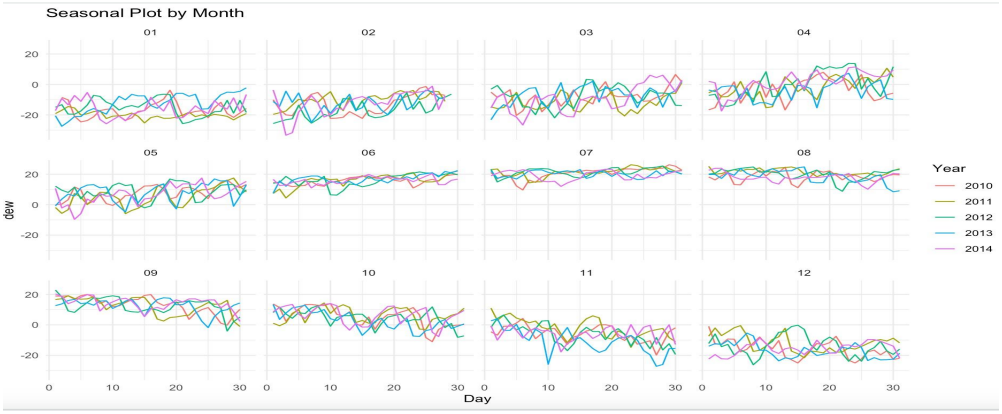


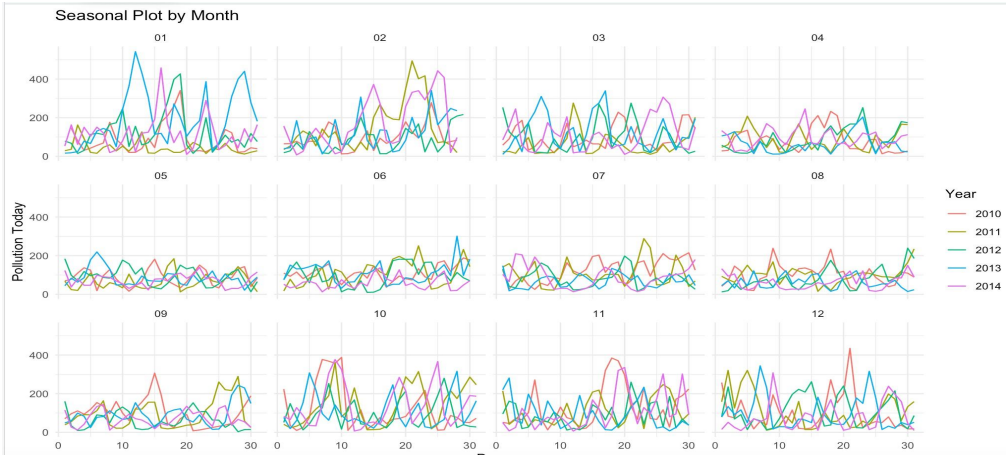
Figure 2.2

Figure 2.2 clearly shows that the AQI value in the early months of 2013 reached the highest point in five years, exceeding 500, during the period from 2010 to 2014. However, after this peak, the AQI value gradually decreased throughout the year. The fluctuation in dew and temperate is highly similar, indicating a strong correlation between them. In contrast, the fluctuation in wind speed is opposite to that of dew and temperature, demonstrating a negative correlation. Precipitation in Beijing is generally low, with only exceeding 10 millimeters in the early months of 2010.

2.3.3 Seasonal Plot







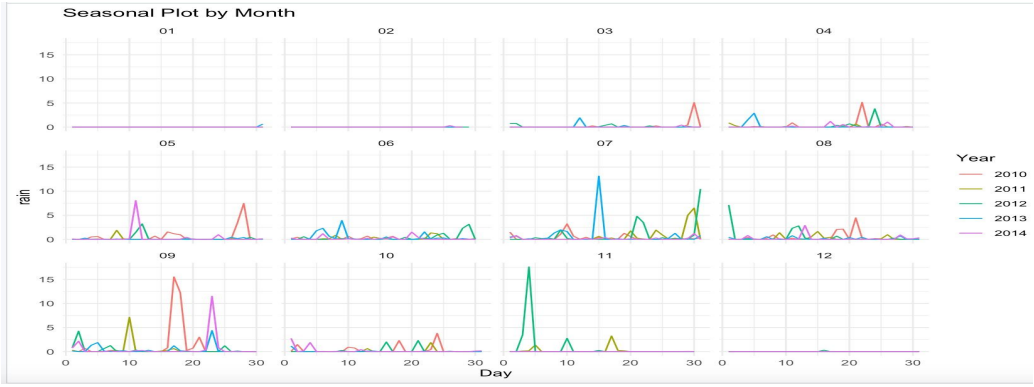
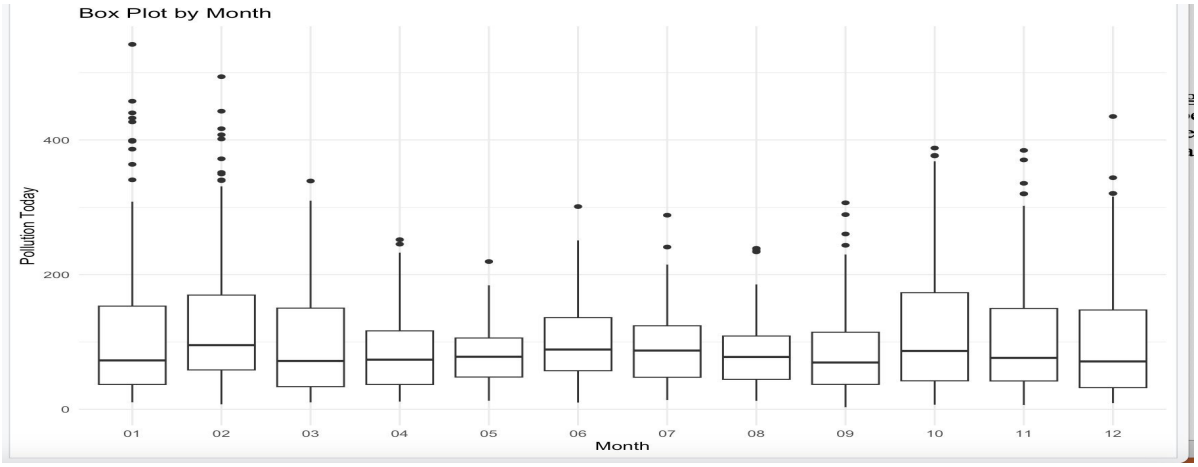


Figure 2.3

Over the five-year period in Figure 2.3, AQI values in October, November, and December were generally higher compared to other months. In contrast, the AQI in May, June, July, and August were relatively stable and remained below 200. Beijing experiences lower precipitation during the winter months, while dew is more prevalent during the summer. In addition, wind speed is lower during the summer and higher during the winter. Therefore, spring is highly susceptible to sandstorms due to increased wind speeds and humidity. During the winter, the increased use of heating in the northern region coupled with low precipitation can easily lead to the formation of haze, which has a severe impact on air quality and public health.

2.3.4 Boxplot



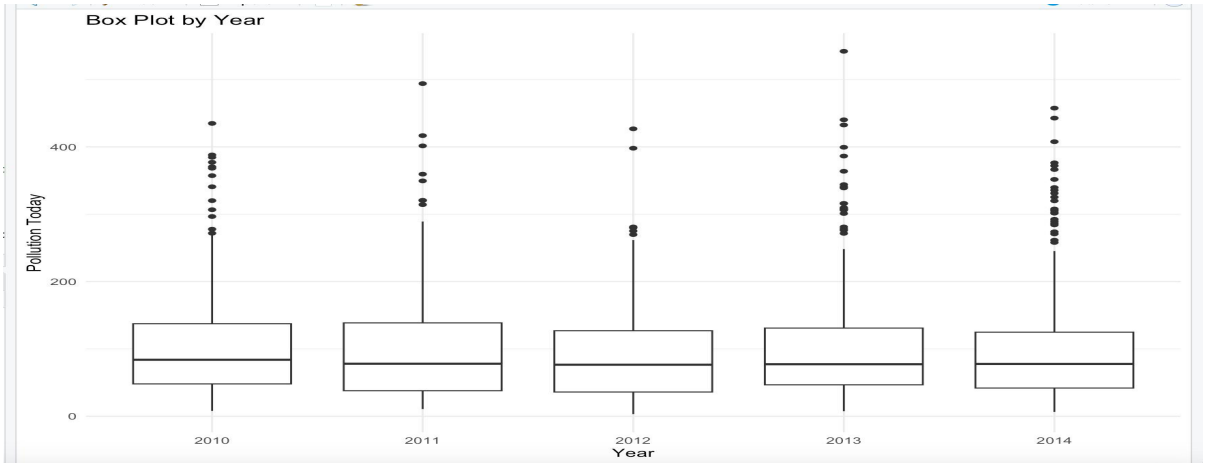


Figure 2.4

From the above box plot in Figure 2.4, we can conclude that there were fewer outliers in 2012, and the overall AQI was more stable compared to other years. The median values for the five years were roughly equivalent. Outliers in the summer were generally fewer than those in the winter. In May, the difference between the lower quartile and the upper quartile was the smallest. The data distribution for each month was positively skewed.

3. Model
3.1 Linear Regression Model Analysis

3.1.1 Model Description

A linear regression model is a type of predictive statistical model used to establish the relationship between a dependent variable (target) and one or more independent variables (features) by fitting a linear equation to observed data. The aim is to find the best-fitting line through the data points that minimizes the sum of the squared differences between the observed values and the predicted values.

The performance of a linear regression model is typically evaluated by R-squared, adjusted R-squared, and root mean squared error (RMSE). R-squared indicates the proportion of the variance in the dependent variable that is predictable from the independent variables, while RMSE measures the average difference between the observed and predicted values of the dependent variable.

The general form of a linear regression equation is :

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n + \varepsilon$$

- Y is the dependent variable (target)
- $X_1, X_2, ..., X_n$ are the independent variables (features)
- β_0 is the intercept (the value of Y when all independent variables are equal to zero)
- $\beta_1, \beta_2, ..., \beta_n$ are the coefficients associated with each independent variable, representing the change in the dependent variable for a one-unit change in the respective independent variable, assuming all other variables remain constant
- ε is the residual error term that accounts for the difference between the actual and predicted values of the dependent variable

In this report, linear regression models will be extended and improved by incorporating regularization techniques (e.g., Ridge and Lasso regression), and feature selection methods. And we will compare the R square and RMSE to choose the best-fitted model.
Below is an explanation of the advantages and disadvantages of linear regression:

Advantages:

- Simplicity and interpretability: Linear regression models are easy to understand and interpret because it only includes linear relationships between dependent and independent variables
- Speed and efficiency: Linear regression models compute efficiently and can be trained quickly, even for a large data set.

- Provides a quantitative relationship: Linear regression models provide a quantitative relationship between the dependent and independent variables, which can be useful for predicting the outcomes.

Disadvantages:

- Assumes linearity: Linear regression models assume that the relationship between the dependent and independent variables is linear. This may not be true for all datasets.
- Sensitive to outliers: Linear regression models can be significantly affected by outliers in the data.
- Multicollinearity: When independent variables in the dataset are highly correlated, it can lead to multicollinearity, which may cause instability in the coefficient estimates and make it difficult to interpret

3.1.2 Data Exploration and Visualization

First, a pair plot is employed to scatter plots of ‘pollution_today’ against seven other variables ('dew', 'temp', 'press', 'wnd_spd', 'snow', 'rain', and 'pollution_yesterday'). The scatter plots are displayed in a 3x3 grid, with each subplot showing the relationship between pollution_today and one of the other variables. From the below graph, it is clearly seen that there shows a clear linear relationship between ‘pollution_today’ and ‘pollution_yesterday’; By examining how closely the points cluster around the line, there is a stronger relationship for ‘dew’, ‘temp’, ’press’ with ‘pollution_today’.

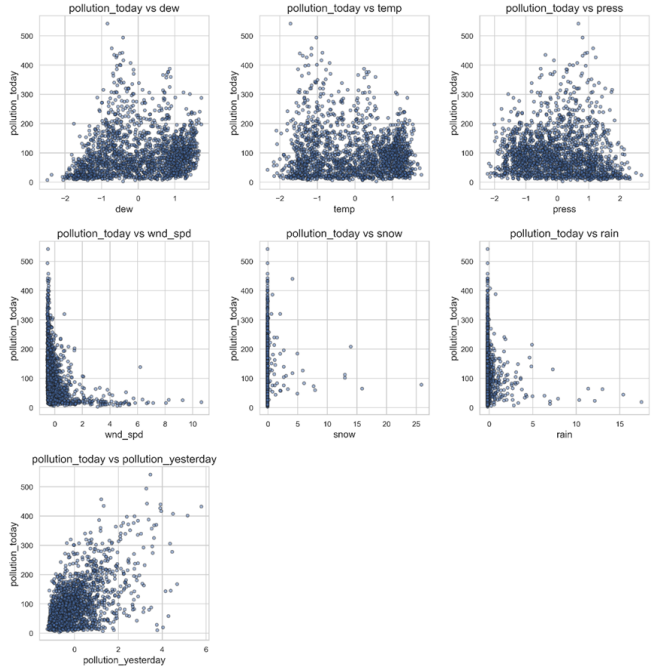


Figure 3.1

By examining the correlation matrix in Figure 3.1, the strongest correlation is between ‘pollution_today’ and ‘pollution_yesterday,’ with a positive correlation coefficient of 0.568. This

indicates that if the pollution level was high yesterday, it is likely to be high today as well. There is a negative correlation between ‘pollution_today’ and ‘wnd_spd’ (-0.285), which suggests that higher wind speeds tend to be associated with lower pollution levels. There is a strong positive correlation between ‘dew’ and ‘temp’ (0.906), as well as a strong negative correlation between ‘temp’ and ‘press’ (-0.865) and between ‘dew’ and ‘press’ (-0.802). These high correlations among independent variables may indicate multicollinearity, which can affect the accuracy and interpretation of a linear regression model.

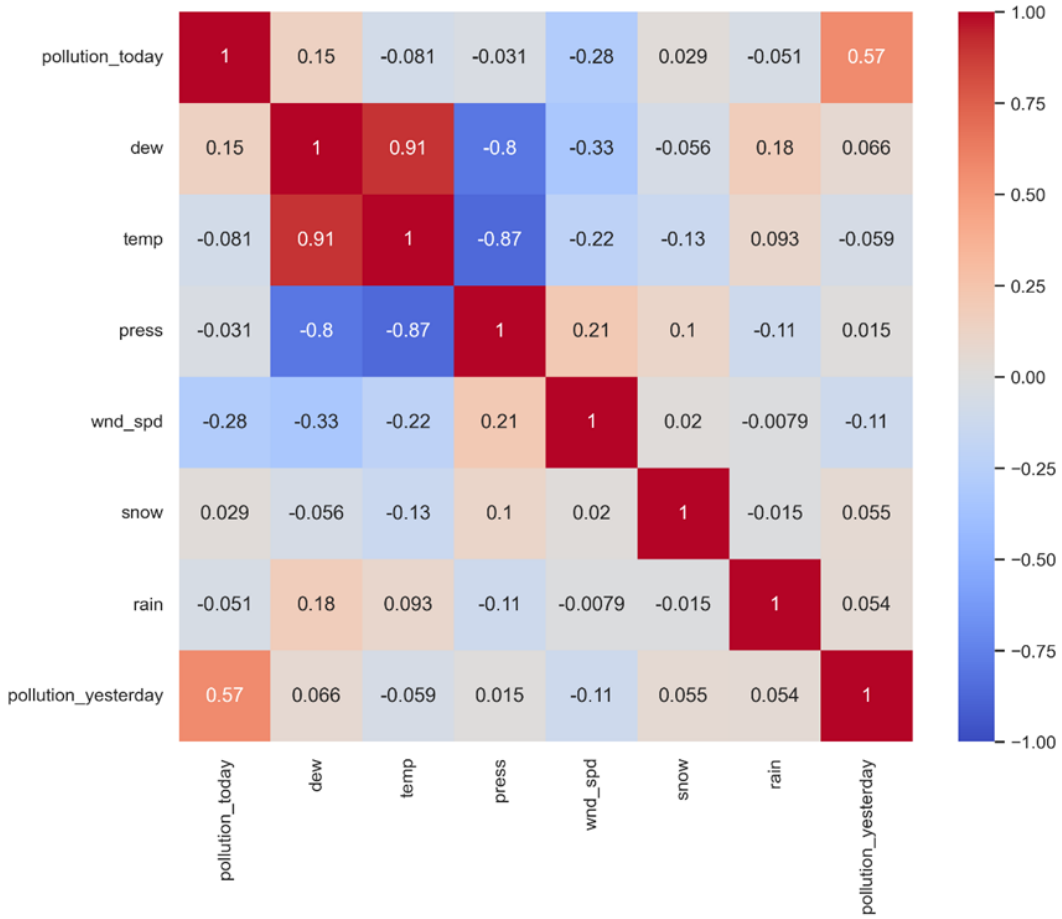


Figure 3.2

3.1.3 Feature Engineering and Models

3.1.3.1 Feature Engineering

First, I tried to standardize the columns excluding the ‘date’ and ‘pollution_today’ columns, using ‘StandardScaler’ from ‘Sklearn. preprocessing’ module. Standardization rescales the features so that they have a mean of 0 and a standard deviation of 1, which can be helpful for further machine learning algorithms. In terms of creating new features, the new feature 'wnd_spd_squared' by squaring the 'wnd_spd' values. This nonlinear transformation can help capture the effect of wind speed on pollution levels in a non-linear manner. Another new feature 'press_log' is created by taking the natural logarithm of the 'press' values. This transformation can

help capture the effect of pressure on pollution levels in a non-linear manner. Moreover, three interaction terms: 'dew_temp', 'dew_press', and 'press_temp' is created by multiplying the respective feature pairs. Interaction terms can help capture the combined effect of two features on the target variable. This step helps improve the model's ability to capture the mode complex relationship between features and target variables.

3.1.3.2 Model A

For model A, which is performed before creating new features. Feature columns are all columns except 'date' and 'pollution_today', while the target column is 'pollution_today'. Data is split into a training set (80% of the data) and a testing set (20% of the data) using the 'train_test_split' function from 'sklearn.model_selection'. Then an Ordinary Least Squares (OLS) linear regression model is fitted on the training data. From the summary of the fitted model, The R-squared value of 0.536 for the training data indicates that the model explains approximately 53.6% of the variability in the 'pollution_today' values. The RMSE of 55.94 indicates the average error between the predicted 'pollution_today' values and the actual 'pollution_today' values in the testing data. The p-values for each feature's coefficient indicate whether the feature has a significant effect on the target variable. All p-values are less than 0.05, which means all features are statistically significant at the 95% confidence level. The Durbin-Watson statistic of 2.025 suggests that there is no significant autocorrelation in the residuals.

OLS Regression Results

Dep. Variable:	pollution_today	R-squared:	0.536
Model:	OLS	Adj. R-squared:	0.534
Method:	Least Squares	F-statistic:	239.5
Date:	Mon, 24 Apr 2023	Prob (F-statistic):	8.39e-237
Time:	00:32:55	Log-Likelihood:	-7830.7
No. Observations:	1460	AIC:	1.568e+04
Df Residuals:	1452	BIC:	1.572e+04
Df Model:	7		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	98.3828	1.356	72.553	0.000	95.723	101.043
dew	70.1931	3.662	19.166	0.000	63.009	77.377
temp	-87.3882	4.169	-20.960	0.000	-95.567	-79.210
press	-22.0971	2.727	-8.102	0.000	-27.447	-16.747
wnd_spd	-8.9549	1.425	-6.286	0.000	-11.749	-6.160
snow	-4.4913	1.367	-3.285	0.001	-7.174	-1.809
rain	-14.2874	1.616	-8.842	0.000	-17.457	-11.118
pollution_yesterday	34.5744	1.436	24.081	0.000	31.758	37.391

Omnibus:	95.266	Durbin-Watson:	2.025
Prob(Omnibus):	0.000	Jarque-Bera (JB):	153.508
Skew:	0.504	Prob(JB):	4.64e-34
Kurtosis:	4.228	Cond. No.	6.74

3.1.3.3 Model B

Model B is performed after creating new features, which incorporate the non-linear transformations and interaction terms into the OLS linear regression model and evaluate its performance using R-squared and RMSE. The R-squared value has increased to 0.541, which indicates that the model with the new features explains about 54.1% of the variance in the target variable 'pollution_today'. This is an improvement over the previous model, which had an R-squared value of 036. The RMSE has also decreased from 55.94 to 54.55, which suggests that the new model has slightly better predictive accuracy.

OLS Regression Results

Dep. Variable:	pollution_today	R-squared:	0.567
Model:	OLS	Adj. R-squared:	0.564
Method:	Least Squares	F-statistic:	158.0
Date:	Mon, 24 Apr 2023	Prob (F-statistic):	7.98e-253
Time:	00:35:12	Log-Likelihood:	-7779.8
No. Observations:	1460	AIC:	1.559e+04
Df Residuals:	1447	BIC:	1.565e+04
Df Model:	12		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	103.6300	2.862	36.213	0.000	98.016	109.244
dew	68.2644	3.790	18.012	0.000	60.830	75.699
temp	-87.4189	4.375	-19.983	0.000	-96.000	-78.838
press	-21.0688	2.746	-7.673	0.000	-26.455	-15.683
wnd_spd	-19.1740	2.731	-7.022	0.000	-24.530	-13.818
snow	-4.6933	1.351	-3.474	0.001	-7.343	-2.043
rain	-14.0195	1.589	-8.825	0.000	-17.136	-10.903
pollution_yesterday	33.6195	1.424	23.617	0.000	30.827	36.412
wnd_spd_squared	2.4493	0.452	5.424	0.000	1.563	3.335
press_log	-3.5552	2.317	-1.534	0.125	-8.101	0.991
dew_temp	-5.6552	3.315	-1.706	0.088	-12.158	0.847
dew_press	11.7410	4.206	2.792	0.005	3.491	19.991
press_temp	-6.1463	3.569	-1.722	0.085	-13.147	0.855

Omnibus:	81.979	Durbin-Watson:	2.041
Prob(Omnibus):	0.000	Jarque-Bera (JB):	136.225
Skew:	0.438	Prob(JB):	2.63e-30
Kurtosis:	4.214	Cond. No.	28.1

3.1.3.4 Model C & D

Recursive Feature Elimination

To begin with, Recursive Feature Elimination with Cross-Validation (RFECV) to identify the optimal number of features for the linear regression model and to select the most important features. The RFECV process identified that the optimal number of features is 13, which includes all the features previously added to the dataset. This means that the algorithm found these 13 features to be important in predicting the target variable 'pollution_today'. Model C & D is performed by applying the optimal features selected by RFECV to fit.

Model C (Ridge Regression)

Model C (Ridge Regression) is using the optimal selected features by RFECV. Ridge Regression (L2 regularization) adds a penalty term to the linear regression objective function, which is proportional to the square of the coefficients. The penalty term is controlled by a hyperparameter, alpha (also denoted as λ), which determines the strength of the regularization. As alpha increases, the coefficients are more heavily penalized, and they shrink toward zero. Ridge regression is a regularization technique that helps improve the generalization of linear regression models. From the results of Ridge regression, the R-square of Ridge regression is 0.5407 and its RMSE is 55.9323.

Model D (Lasso Regression)

Model C (Lasso Regression) is using the optimal selected features by RFECV. Lasso Regression (L1 regularization) adds a penalty term to the linear regression objective function. However, the penalty term in Lasso regression is proportional to the absolute values of the coefficients (excluding the intercept). Similar to Ridge regression, the strength of the penalty is controlled by a hyperparameter, alpha (also denoted as λ). However, different from Ridge Regression, it enforces sparsity and can perform feature selection. From the results of the Lasso regression, the R-square of the Lasso regression is 0.5396 and its RMSE is 55.9406.

3.1.4 Model Selection

Comparing Models A and B, Model A has an R-square of 0.536 and an RMSE of 55.94. After incorporating the non-linear transformations and interaction terms into the OLS linear regression model, we get model B. Model B has a better performance for both R-square(0.567) and RMSE(54.55). Then we try to compare models B, C, and D, the performance of all three models is quite similar, with RMSE values close to each other. The Linear Regression model has an RMSE of 55.9408, the Ridge Regression model has an RMSE of 55.9323, and the Lasso Regression model has an RMSE of 55.9406. These results suggest that the optimal features selected by RFECV provide similar predictive performance for all three models. From the RMSE, we prefer to choose Ridge Regression whose RMSE is 55.9323**,** indicating the average error between the predicted ‘pollution_today’ values and the actual ‘pollution_today’ values in the testing data.

3.2 Time Series Analysis

Time series models are models used to predict future data points, typically based on past time series data to infer future data. Common models used in time series modeling include LSTM, AR, MA, ARMA, ARIMA, and SARIMA. Below we will outline the differences, advantages and disadvantages, and applicable scenarios of these models.

3.2.1 Model Description

A. ARIMA Model

The Autoregressive Integrated Moving Average (ARIMA) model is an extension of the ARMA model, adding an integration term to handle non-stationary time series. The d in the ARIMA model refers to the number of times the data is differenced, and p and q represent the orders of AR and MA, respectively. The advantages of the ARIMA model are its adaptability to trends and seasonal changes in time series data, as well as its ability to handle nonlinear data. However, parameter selection for the ARIMA model requires experience and time, and the time series data must be stable. When the data lacks stability, differencing is needed to make the data stable, but this will increase the complexity of the model. In addition, the ARIMA model requires many assumptions, such as the residuals of the data should be white noise, and the sample size should be sufficient.

AR, MA, and ARMA are subsets of ARIMA, which are used to build autoregressive, moving average, and autoregressive moving average models, respectively. These models are typically used to predict stable time series data, and their advantages are relatively simple parameter selection, and they are usually easier to understand and interpret than the ARIMA model. However, the prediction ability of these models is limited by their assumptions. If the data does not meet these assumptions, then their prediction performance will be unsatisfactory.

B. SARIMA Model

Seasonal Autoregressive Integrated Moving Average, SARIMA or Seasonal ARIMA, is an extension of ARIMA that explicitly supports univariate time series data with a seasonal component. It adds three new hyperparameters to specify the autoregression (AR), differencing (I), and moving average (MA) for the seasonal component of the series, as well as an additional parameter for the period of the seasonality (Brownlee, 2018).

C. LSTM Model

The LSTM model is a type of recurrent neural network model that can capture more complex patterns and trends in time series data. Compared to traditional models, the LSTM model does not require data to be stationary and can model and predict directly using raw time series data. At the same time, the LSTM model can also capture long-term dependencies in the data, thereby improving prediction performance.

Advantages:

- Can capture more complex patterns and trends in time series data.
- Does not require data to be stationary.
- Can handle long-term dependencies, improving prediction performance.

Disadvantages:

- Requires more computing resources and time due to its large computational complexity.

- May overfit simple time series data.

Applicable scenarios:

- Handling complex time series data.
- Handling time series data with long-term dependencies.

The advantage of the LSTM model is that it can capture long-term dependencies in time series data, making it suitable for modeling and predicting nonlinear and non-stationary time series data. The LSTM model does not require data to be differenced, so it can handle non-stationary time series data. However, the LSTM model has a longer training time and requires significant computing resources. In addition, parameter tuning for the LSTM model also requires experience and time, and the risk of overfitting is relatively high.
We will use these time series models to analyze the data and predict the future performance. Meanwhile, we also want to compare these models to get the best model performance.

3.2.2 ARIMA

Firstly, we need to import the necessary Python libraries, including Pandas, Matplotlib, and Statsmodels. These libraries can help us with data processing, visualization, and time series analysis. Next, we can read the dataset, convert it to a Pandas DataFrame, and set the dates as the index. Before performing a time series analysis on the dataset, we need to do some data preprocessing. Firstly, we can check if the dataset contains missing values and fill them using interpolation methods. Then, we can plot the graph of the time series to understand its features and trends better. From the output above, it can be seen that the dataset contains 1825 data points, i.e., hourly data for one year. The average value of the pollutant concentration is 98, the standard deviation is 77, the minimum value is 3.1, and the maximum value is 542.

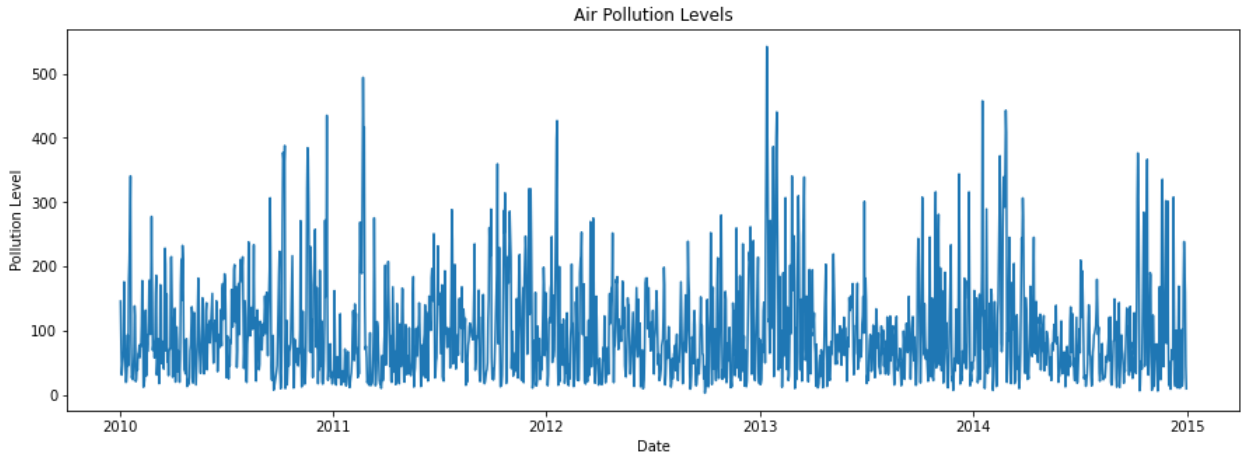


Figure 3.3

Next, we can use time series analysis methods to explore the trend, seasonality, and noise features of the data. We will use the statsmodels library to perform these analyses.

Firstly, we will use the *seasonal_decompose()* function of the statsmodels library to decompose the data into trend, seasonality, and noise components. This function uses the STL (Seasonal-Trend decomposition using Loess) method, which is a commonly used decomposition method based on local weighted regression (LOWESS).

This segment will decompose the data into trend, seasonality, and noise components and plot the decomposed graph.

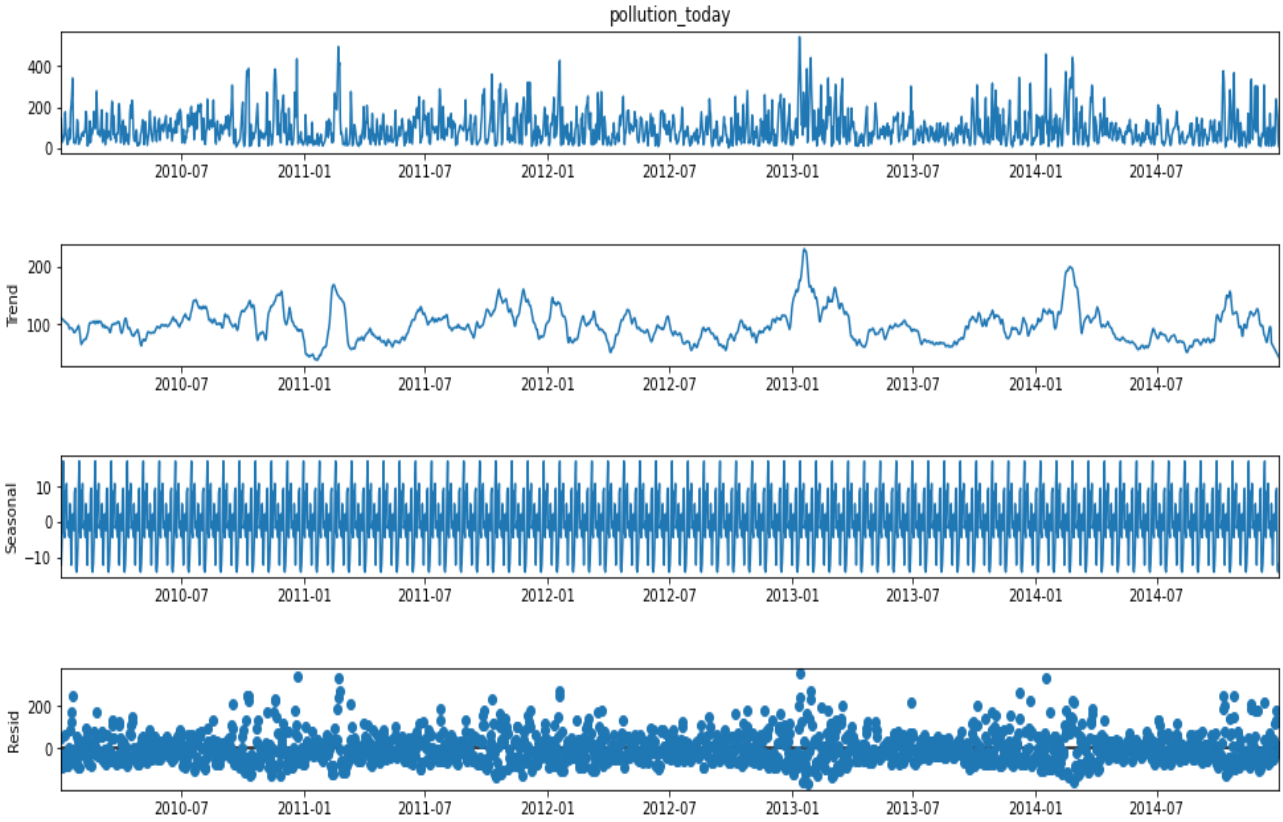


Figure 3.4

From Figure 3.4, it can be observed that the time series of pollution concentration has significant seasonality and trend. The seasonality changes with time, while the trend appears to be linear.

Next, we will use the *autocorrelation function (ACF)* and *partial autocorrelation function (PACF)* to determine the order of the time series and the hyperparameters of the ARIMA model.

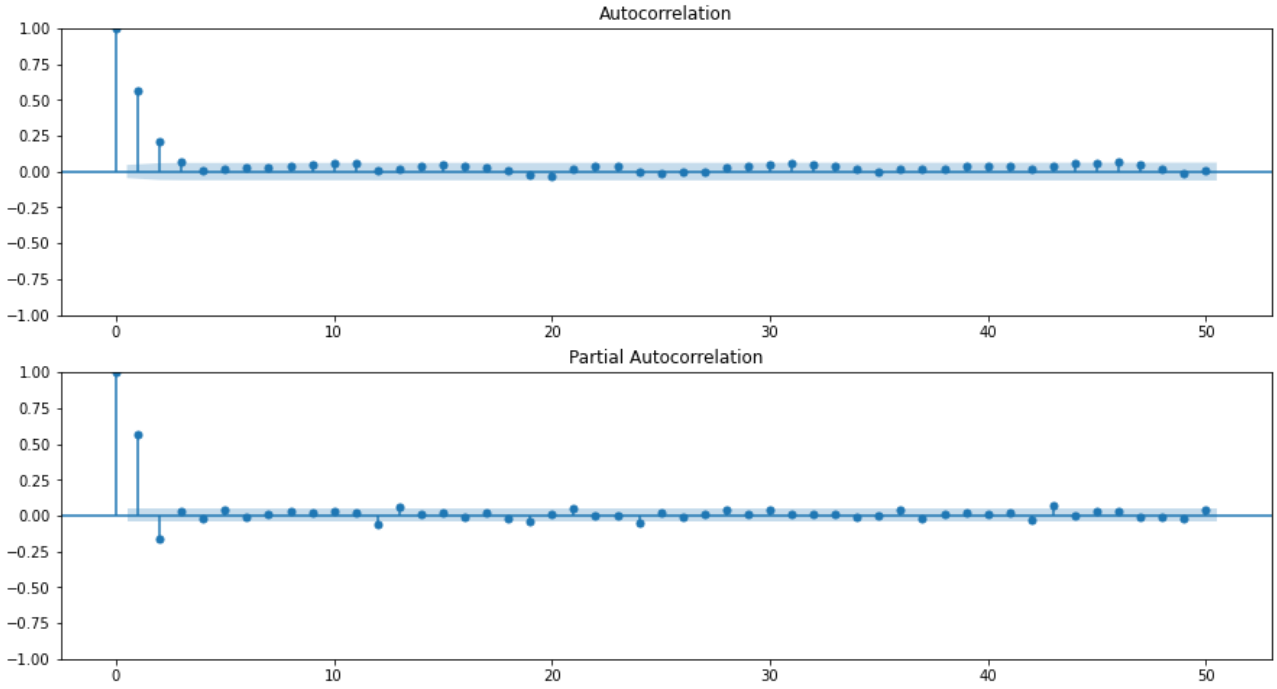


Figure 3.5

In Figure 3.5, we can see that the time series of pollutant concentration has significant seasonality and trend, where seasonality changes with time, and trend appears to be linear.

Next, we will use the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) to determine the order of the time series and hyperparameters of the ARIMA model.

As we can see from the figure, ACF quickly drops to zero, indicating that we can try to use the Moving Average (MA) model to fit the data. And PACF also drops to zero quickly, indicating that we can try to use the Autoregressive (AR) model to fit the data. Therefore, we can try to use the ARMA model to fit the data.

Next, we will use the ARIMA class from the statsmodels library to fit the ARIMA model. The order of the ARIMA model needs to be selected based on the ACF and PACF plots. Here, we select the ARIMA(1, 1, 1) model.

SARIMAX Results

Dep. Variable:	pollution_today	No. Observations:	1825
Model:	ARIMA(1, 1, 1)	Log Likelihood	-10152.851
Date:	Sun, 23 Apr 2023	AIC	20311.701
Time:	16:53:58	BIC	20328.228
Sample:	01-02-2010 - 12-31-2014	HQIC	20317.798
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.5699	0.012	46.478	0.000	0.546	0.594
ma.L1	-1.0000	0.082	-12.214	0.000	-1.160	-0.839
sigma2	3988.8831	347.997	11.462	0.000	3306.822	4670.944

Ljung-Box (L1) (Q):	15.94	Jarque-Bera (JB):	398.84
Prob(Q):	0.00	Prob(JB):	0.00
Heteroskedasticity (H):	1.12	Skew:	0.62
Prob(H) (two-sided):	0.17	Kurtosis:	4.92

Figure 3.6

In Figure 3.6, we can see that the AIC value of the ARIMA(1, 1, 1) model is 20311.7 and the BIC value is 20328.228. Meanwhile, the AR coefficient of the model is 0.5699, the MA coefficient is -1, and the noise variance is 3988.8831.

We can also use the *plot_diagnostics()* function in the statsmodels library to plot the diagnostic graphics of the ARIMA model for further evaluation. (Above)

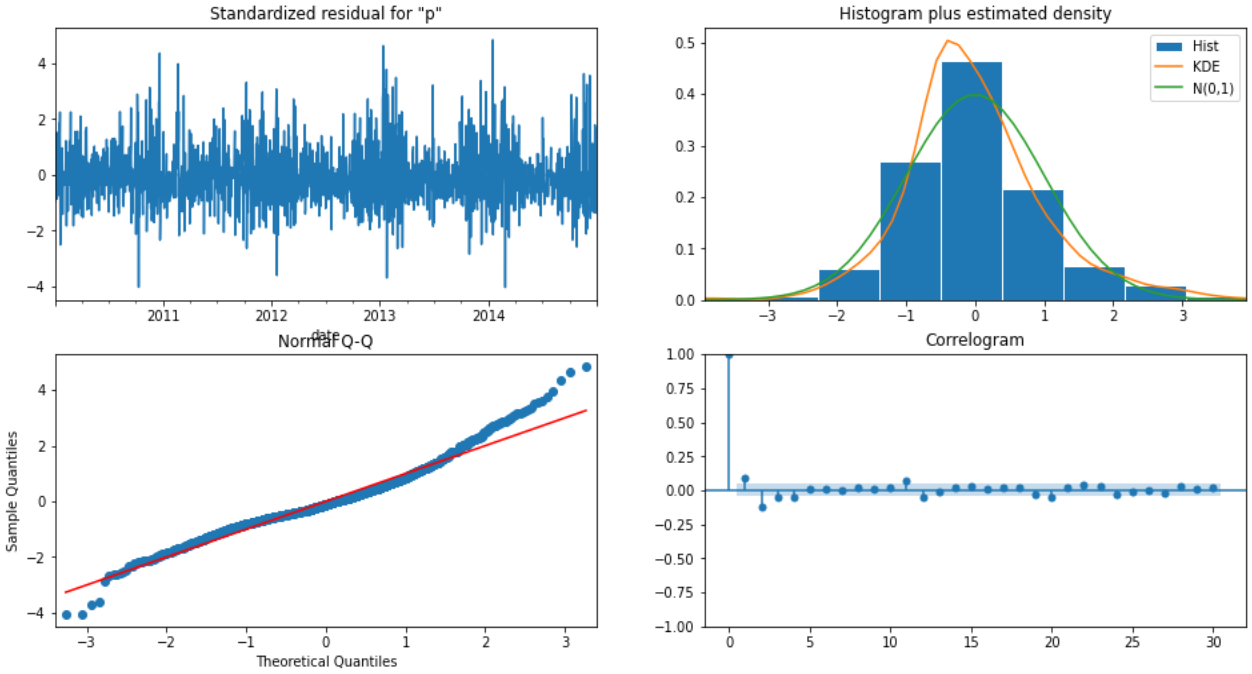


Figure 3.7

In Figure 3.7, it can be seen that the ACF and PACF of the standardized residual sequence fluctuate around zero and are not significant at all lag orders. This indicates that the

ARIMA(1, 1, 1) model performs well in terms of autocorrelation and partial autocorrelation of the residual sequence, with no missed autocorrelation structures. Additionally, the histogram and normal Q-Q plot of the residual sequence show a relatively normal distribution, with no significant skewness or heavy-tailedness issues.

In conclusion, the ARIMA(1, 1, 1) model fits well with the pollution_today variable in the air pollution dataset, and the residual sequence of the model shows a relatively normal distribution. Now, we can use this model to predict future pollution indices. (Above)

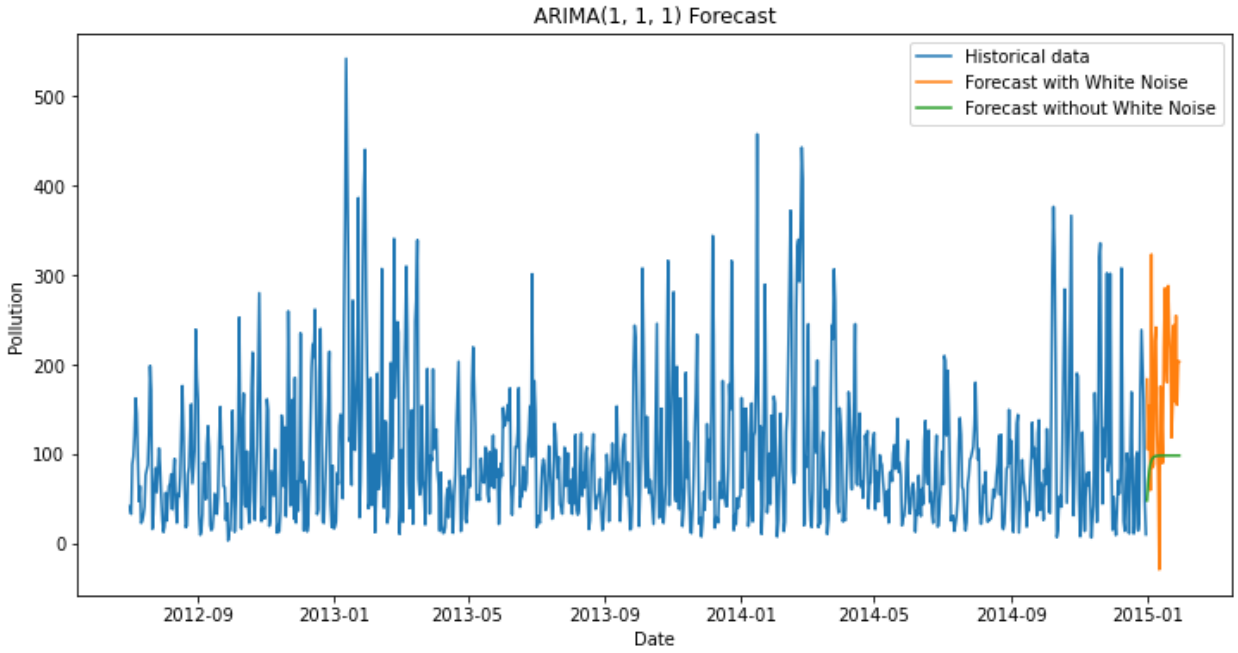


Figure 3.8

As shown in Figure 3.8, the forecast of the ARIMA(1, 1, 1) model is consistent with historical data, and the future pollution index is expected to exhibit a stable trend.

Although the method can handle data with a trend, it does not support time series with a seasonal component. We choose to use SARIMA, which is an extension of ARIMA that supports the direct modeling of the seasonal component (Brownlee, 2018).

3.2.3 SARIMA

3.2.3.1. Data Inspection and Model Choosing

In order to make data more feasible to do time series analysis, we first convert the daily pollution data into monthly pollution data. From the monthly data, we leave 12 months, which is one year in order to test whether the prediction is right.

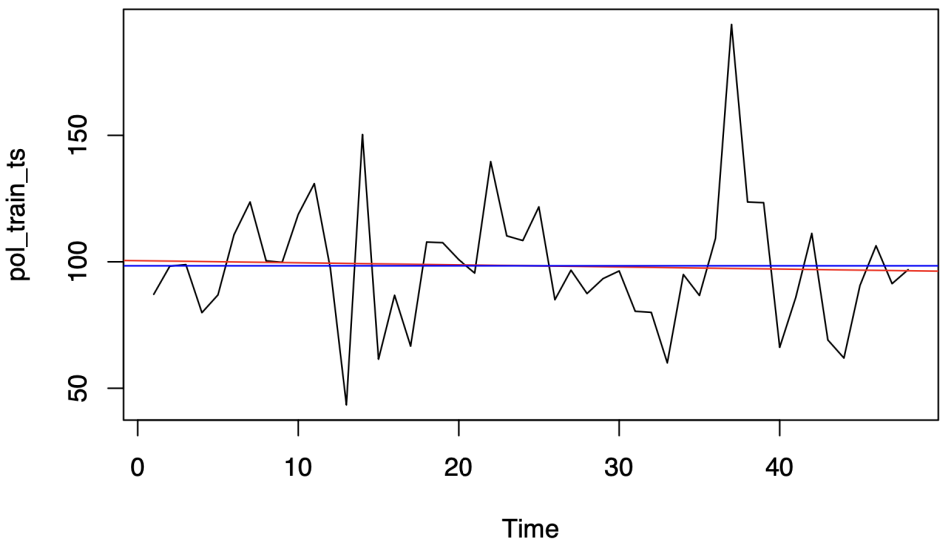


Figure 3.9

In Figure 3.9, the red line in the plot represents the linear regression line, which is the best-fitting straight line through the data points. The blue line represents the mean of the time series. As both lines overlap in the plot, it implies that the mean of the time series and the linear regression line are very close or nearly the same.

It suggests that the linear trend in the time series data is essentially flat. This means that, on average, the monthly pollution level has not changed significantly over time. However, a linear regression might not always be the best model to capture the underlying trend in the time series data, especially if the data has a more complex, non-linear pattern. Thus, we are going to try other models.

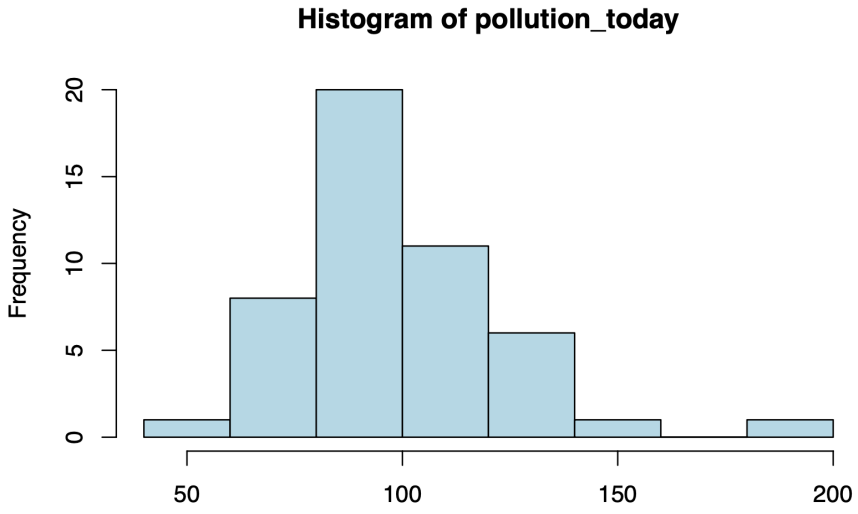


Figure 3.10

From Figure 3.10 the histogram of the data, we can tell the data is skewed, which means the variance is non-constant (heteroscedasticity). This is problematic for time series analysis. In order to have a constant variance for further steps, we can try the Box-Cox transformation.

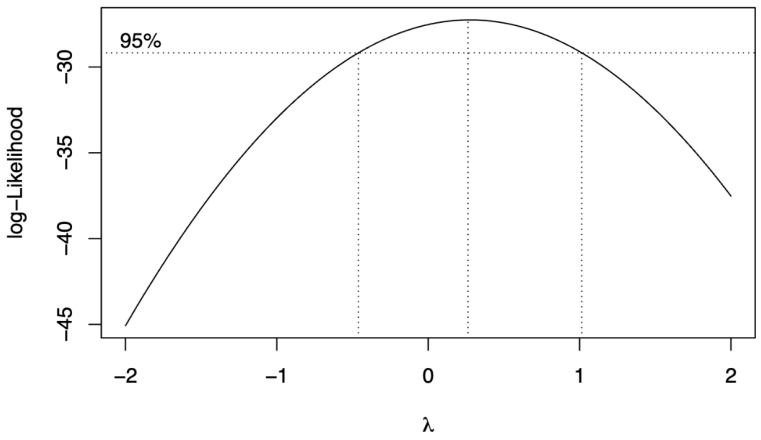


Figure 3.11

From the result in Figure 3.11, $\lambda = 0.26$, but since 0 is also in the confidence interval, for simplicity, we will just use $\lambda = 0$, which is the log transformation.

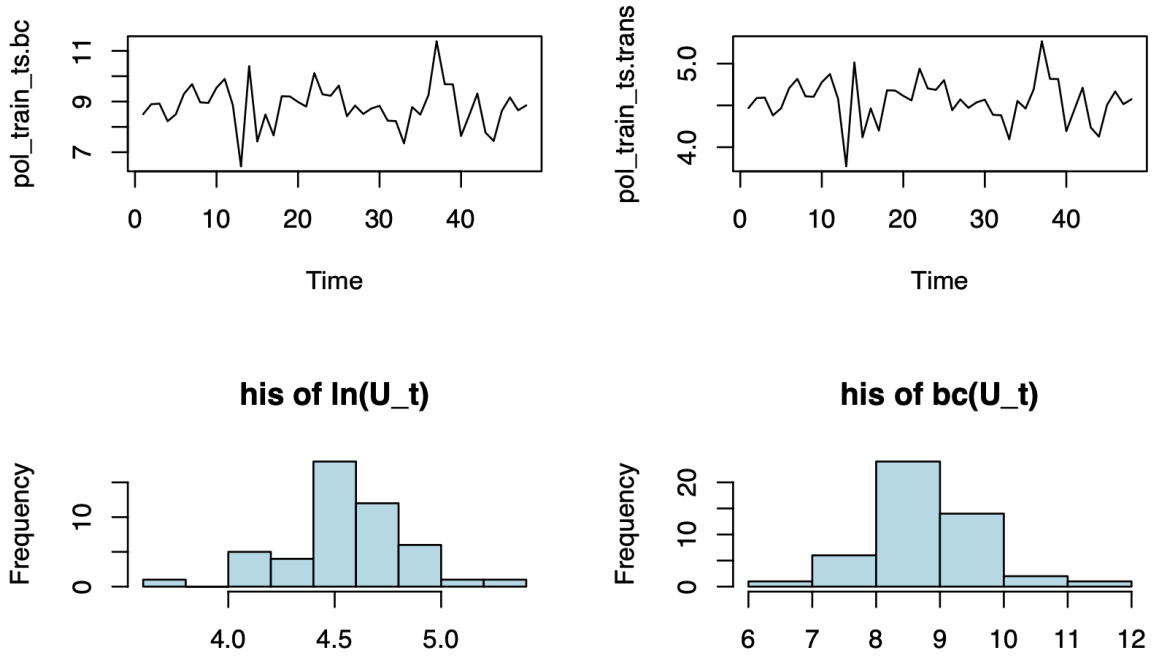


Figure 3.12

In Figure 3.12, the right histogram of log transformation looks similar to the left histogram of Box-Cox, which illustrates the data transformed is normally distributed and that we stabilize the variance.

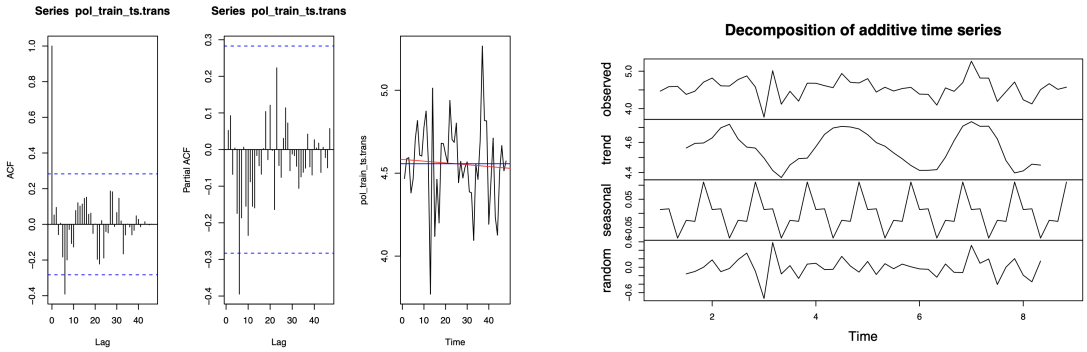


Figure 3.13

The decomposition in Figure 3.13, shows that our data exists in seasonality. From the ACF graph, there is one line at lag 6 that is noticeable, which indicates the seasonality component is 6, so the next step is to differentiate the data once at lag 6 to remove the seasonality.

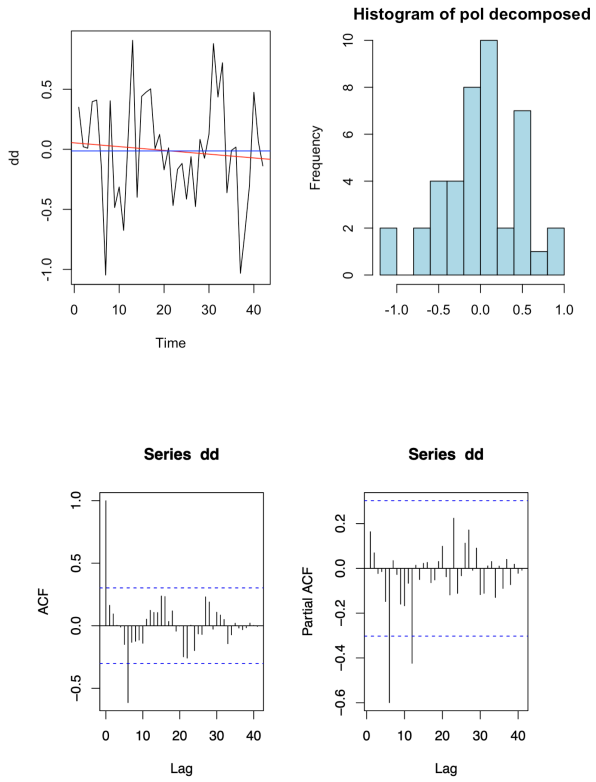


Figure 3.14

After decomposing the seasonality once, we still can see a noticeable peak at lag 6 in ACF from Figure 3.14. There is still some remaining seasonality in the data even after the first seasonal differencing. In the PACF plot, noticeable peaks at lags 6 and 12 suggest that there is some remaining autocorrelation in the data that is not explained by the seasonal differencing. To address the remaining autocorrelation and seasonality in the data, we considered using a Seasonal ARIMA (SARIMA) model with a period of 6, which can simultaneously model the non-seasonal AR, non-seasonal MA, seasonal AR, and seasonal MA components.

	p	q	P	Q	AICc
19	0	0	2	0	21.13268
37	0	0	1	1	21.69124
20	1	0	2	0	22.71904
22	0	1	2	0	22.79455
46	0	0	2	1	23.01427
38	1	0	1	1	23.50245

Figure 3.15

In order to choose an appropriate model, we ran a for loop to estimate which model produced the lowest AICc value as in Figure 3.15. I chose models that have the lowest and the second lowest to estimate the coefficients. From this, we chose our two models:

$Model A : SARIMA(0,0,0)(2,1,0)_6$

$Model B : SARIMA(0,0,0)(1,1,1)_6$

3.2.3.2 Model Diagnose

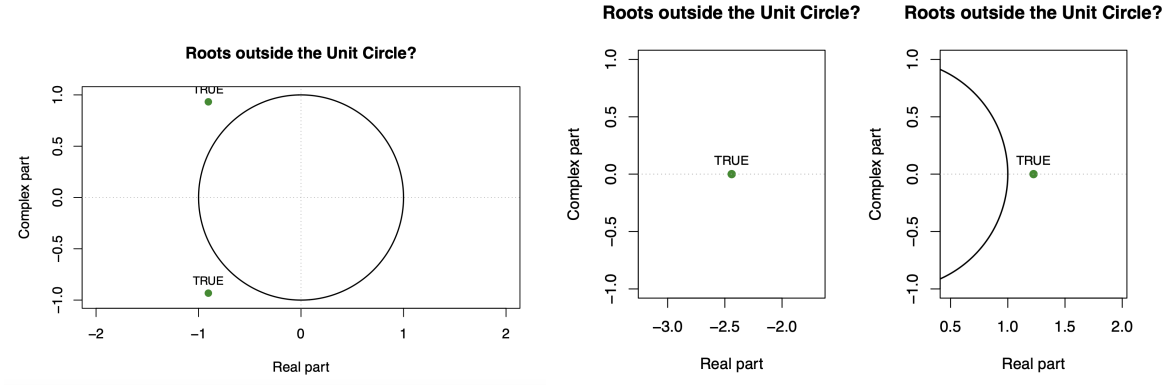
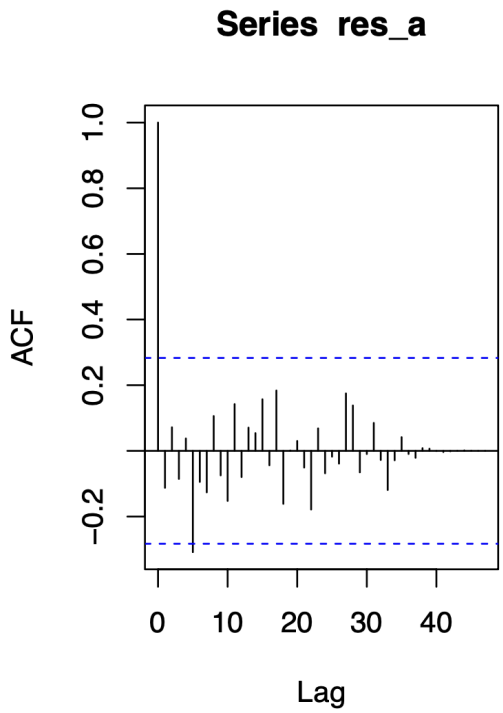
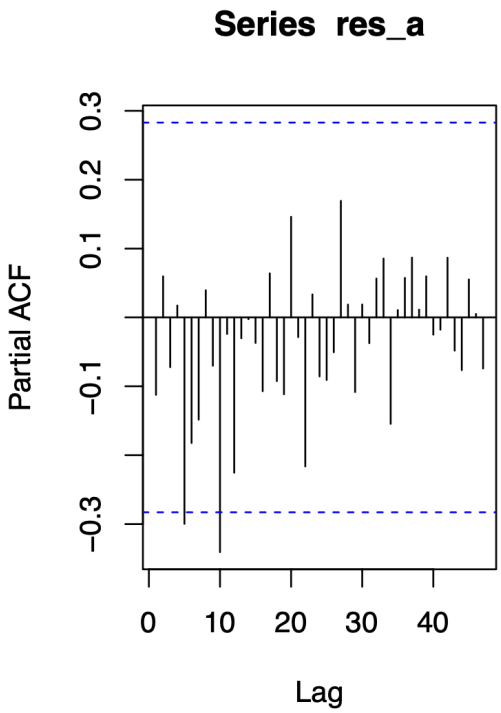


Figure 3.16

We used a unit circle to check the properties of invertibility and causality. From Figure 3.16, we can tell both two models are invertible and causal after estimating the parameters for both models.

a. Model A checking






```
shapiro.test(res_a)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res_a
## W = 0.96092, p-value = 0.11
```

```
Box.test(res_a, lag=7, type=c("Box-Pierce"), fitdf=2)
```

```
##
##  Box-Pierce test
##
## data:  res_a
## X-squared = 7.0266, df = 5, p-value = 0.2187
```

```
Box.test(res_a, lag=7, type=c("Ljung-Box"), fitdf=2)
```

```
##
##  Box-Ljung test
##
## data:  res_a
## X-squared = 8.1271, df = 5, p-value = 0.1494
```

```
Box.test((res_a)^2, lag=7, type=c("Ljung-Box"), fitdf=0)
```

```
##
##  Box-Ljung test
##
## data:  (res_a)^2
## X-squared = 2.1241, df = 7, p-value = 0.9527
```

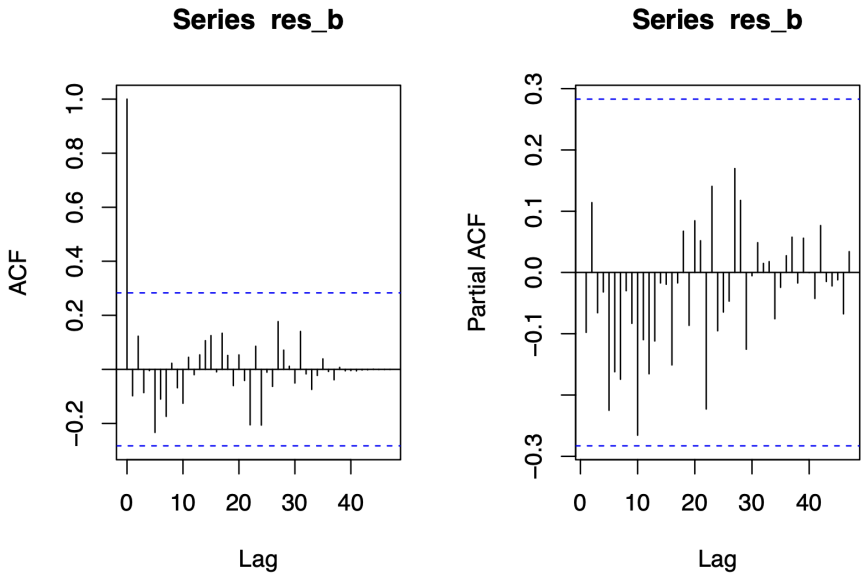
```
ar(res_a, aic=TRUE, order.max=NULL, method=c("yule-walker"))
```

```
##
## Call:
## ar(x = res_a, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  0.06038
```

Figure 3.17

In Figure 3.17, the PACF graph for the residual has some lags that are outside of the confidence interval, which indicates that the residuals are not behaving like a white noise process. Model A may not be a good choice even though it passes all five diagnostic criteria with all p-values larger than 0.05.

b. Model B checking




```
shapiro.test(res_b)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res_b
## W = 0.93859, p-value = 0.01433
```

```
Box.test(res_b, lag=7, type=c("Box-Pierce"), fitdf=2)
```

```
##
##  Box-Pierce test
##
## data:  res_b
## X-squared = 6.1747, df = 5, p-value = 0.2896
```

```
Box.test(res_b, lag=7, type=c("Ljung-Box"), fitdf=2)
```

```
##
##  Box-Ljung test
##
## data:  res_b
## X-squared = 7.1597, df = 5, p-value = 0.209
```

```
Box.test((res_b)^2, lag=7, type=c("Ljung-Box"), fitdf=0)
```

```
##
##  Box-Ljung test
##
## data:  (res_b)^2
## X-squared = 2.5219, df = 7, p-value = 0.9254
```

```
ar(res_b, aic=TRUE, order.max=NULL, method=c("yule-walker"))
```

```
##
## Call:
## ar(x = res_b, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  0.0582
```

Figure 3.18

From the test results of Figure 3.18, although Model B didn't pass the Shapiro-Wilk normality test, it doesn't mean Model B is impossible to use.

Since the residual of Model A is not White Noise, Model B is a better choice here.

3.2.3.3. Forecasting

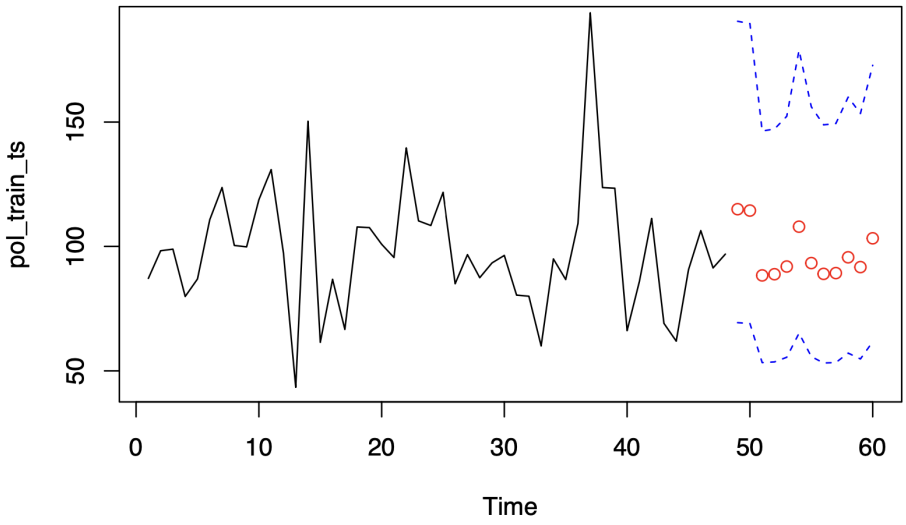


Figure 3.19

Since the data used here is the transformed data, we should transform it back to the original data, and compare the predicted results with the original data. In Figure 3.19, the dashed blue line represents the confidence interval of prediction, and the red points represent the prediction for 12 months. There might be a small peak of AQI after 6 months. After a decrease, the air pollution might increase to another peak again.

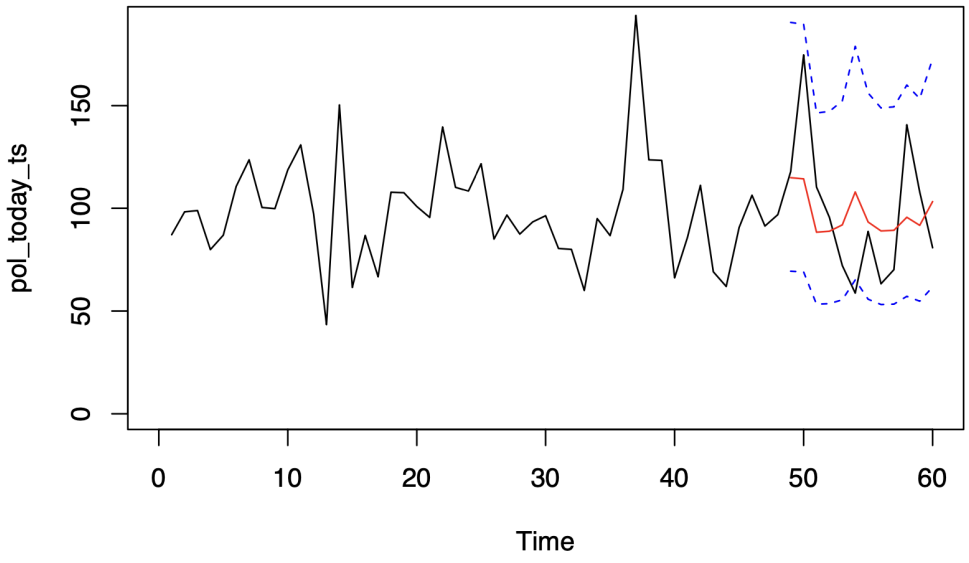


Figure 3.20

In Figure 3.20, the red line represents the prediction for 12 months. We can see that the overall trend is similar to the original data. There are two peaks in the last 12 months, which are the same as the prediction.

```
rmse_value <- rmse(pred.ori, pol_test_ts)
rmse_value
```

```
## [1] 30.04416
```

Figure 3.21

From Figure 3.21, we calculated the MSE for Model B prediction, **30.04416**. Therefore, the final model is $SARIMA(0,0,0)(1,1,1)_6$

3.2.4 LSTM

To use the LSTM model for predicting the pollution_today variable in the air pollution dataset, we need to preprocess the dataset. First, we normalize the values of the pollution_today variable to better handle the data.

Next, we use the sliding window method to transform the dataset into a form suitable for the LSTM model. Here we use a window size of 30 days, meaning we take the past 30 days of pollution index as input features to predict the next day's pollution index.

Now we can define the LSTM model. We use a neural network consisting of two LSTM layers and one fully connected layer, with each LSTM layer containing 64 neurons. To prevent overfitting, we add a Dropout layer after each LSTM layer. The model's loss function is a mean squared error (MSE), and the optimizer is Adam.

After training, we can use the model to predict future pollution index. For convenience, we merge the test and train data and take the last 30 days of data as input to the model. Here are the plots for the training loss and validation loss. From these plots, we can determine the performance of this model.

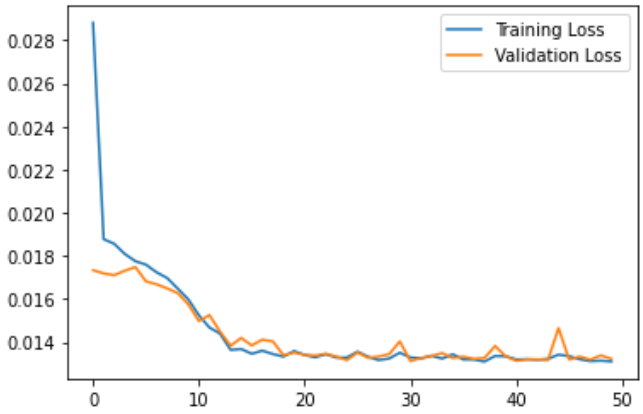


Figure 3.22

In Figure 3.22, the blue line represents the training loss, and the orange line represents the validation loss. We can see that both the training and validation losses are gradually decreasing, indicating that the model is learning and improving its predictions. However, we also need to note that at certain points, the validation loss may increase, indicating that the model may be overfitting the training data. If this happens, we can avoid overfitting by adjusting the model architecture or adding regularization.

After we compile the model, we can predict the air pollution of validation data, which is shown below.

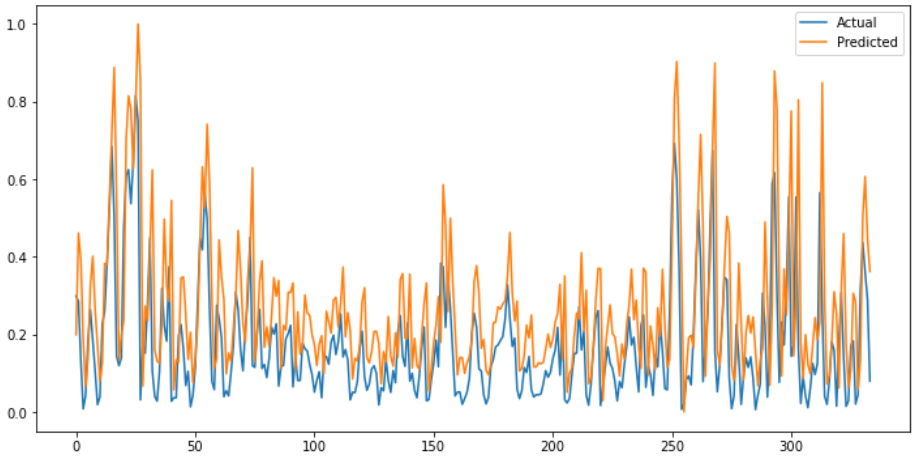


Figure 3.23

In Figure 3.23, the blue line represents the actual values, while the orange line represents the predicted values of the LSTM model. We can see that the LSTM model performs very well in predicting the pollution values for most of the time points, but there are some moments where errors exist. These errors may be due to sudden events or other factors.

4. Conclusion

Model	RMSE
Ridge Regression	55.932
ARIMA(1,1,1)	32.501
$SARIMA(0,0,0)(1,1,1)_6$	30.044
LSTM	12.312

Figure 4.1

In conclusion, our project aimed to predict AQI values using different models and compare their accuracy. We used five years of historical data and applied several models to make predictions. In Figure 4.1, RMSE for LSTM is 12.312 which is the lowest. Thus, we choose LSTM to predict AQI, which handles complex time series data. Additionally, the comparison of different models and their performance provides valuable insights for future research in AQI prediction. From the RMSE of those four models we perform, we can tell there is no clear trend that allows us to predict. Since the AQI values vary a lot, we cannot conclude that there is a clear pattern or trend to it.

References

Brownlee, J. (2018, August 17). *A Gentle Introduction to SARIMA for Time Series Forecasting in Python*. Machinelearningmastery.com. https://machinelearningmastery.com/sarima-for-time-series-forecasting-in-python/

Chen, S. X. (2017, January 19). *Beijing PM2.5 Data Data Set*. UCI Machine Learning Repository: Beijing PM2.5 Data Data Set. Retrieved April 30, 2023, from https://archive.ics.uci.edu/ml/datasets/Beijing+PM2.5+Data#

Huertas, J. F. (2022, September 16). *Time-Series-forecasting-with-python: A use-case focused tutorial for time series forecasting with python.* GitHub.com. https://github.com/jiwidi/time-series-forecasting-with-python